REMARKS

Applicant respectfully requests that the above-identified application be reexamined.

The Office Action mailed on July 1, 2004 ("Office Action"), rejected Claims 1-12, and withdrew Claims 13-22 from consideration.

During a telephone conversation with the Examiner on June 21, 2004, applicant made a provisional election without traverse to prosecute Claims 1-12. Applicant affirms this election and cancels the non-elected Claims 13-22 for the purpose of this application.

The Office Action rejected Claims 6, 7, 11, and 12 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter that applicant regards as the invention.

The Office Action further made multiple rejections under 35 U.S.C. § 103(a). The Office Action rejected Claims 1, 6, and 7 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,664,190, to Cohen et al. (hereinafter "Cohen"), in view of U.S. Patent No. 6,487,652 B1, to Gomes et al. (hereinafter "Gomes"). The Office Action rejected Claims 2-5 under 35 U.S.C. § 103(a) as being unpatentable over Cohen in view of Gomes as applied to Claim 1, and further in view of U.S. Patent No. 5,434,975, to Allen (hereinafter "Allen"). The Office Action rejected Claims 8, 9, 11, and 12 under 35 U.S.C. § 103(a) as being unpatentable over Cohen in view of Allen. The Office Action rejected Claim 10 under 35 U.S.C. § 103(a) as being unpatentable over Cohen in view of Allen as applied to Claim 8, and further in view of Gomes.

Pursuant to 37 C.F.R. § 1.111, and for the reasons herein set forth, applicant respectfully requests reconsideration and allowance of this application.

Prior to discussing the reasons why applicant believes that the claims of this application are clearly allowable, a brief discussion of the present invention, followed by a brief discussion of the cited and applied references, is presented. The following discussions of applicant's invention and the cited and applied references are not provided to define the scope or interpretation of any of the claims of this application. Instead, these discussions are provided to help the United States Patent and Trademark Office better appreciate important claim distinctions discussed thereafter.

Summary of the Invention

The present invention provides an apparatus for providing and integrating high-performance message queues in a user interface environment. Generally described, the present invention provides high-performance message queues in a user interface environment that can

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

scale when more processors are added. This infrastructure provides the ability for user interface components to run independently of each other in separate "contexts."

More specifically described, the present invention provides contexts that allow independent "worlds" to be created and executed in parallel. A context is created with one or more threads. Each object is created with context affinity, which allows only threads associated with the context to modify the object and process pending messages. Threads associated with another context are unable to modify the object or process pending messages for that context.

One aspect of the invention provides a method for sending a message via a high-performance message queue. The method includes providing a message queue associated with a context and executing a user interface thread associated with this context. The method also includes receiving a request from the user interface thread to send a message to a second user interface thread. The method further includes, upon determining that the second user interface thread is associated with the same context, sending the message from the first user interface thread directly to the second user interface thread, bypassing the message queue.

Another aspect of the invention teaches posting a message via a high-performance message queue. The method includes providing a message queue associated with a context, executing a user interface thread associated with the context, and receiving a request from the user interface thread to post the message to a queue associated with a second context. The method also includes atomically adding the message to the queue associated with the second context, and atomically providing an indication to the second context that a message has been posted to the queue associated with the second context.

In summary, the present invention provides the ability for user interface components to run independently of each other in separate "contexts" and still be able to communicate with each other using a high-performance message queue. In practice, this allows communication between different components at a rate of ten to one hundred times faster than what was possible in previous solutions.

Summaries of Cited References

Summary of Cohen

Cohen purportedly provides a system and a method for making procedures written in procedural languages to respond reactively to messages from event-driven interfaces. It aims to provide an interface to enable window procedures written in procedural languages to respond reactively to input from a user at a user interface.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

Cohen teaches a system in a computing environment for enabling procedural programs to respond reactively to messages. The system comprises receiving means for receiving input from the user interface and an operating system for receiving the input from the receiving means. The system also comprises a plurality of message queues for receiving messages from the operating system, one message queue corresponding to each procedural program accessible by the user interface. The system further comprises a procedural program interface for determining whether a message is present in one of the message queues and providing the message to the procedural program following receipt of a call from the procedural program. The procedural program interface may include one procedural program message queue corresponding to each of the message queues. The procedural program interface may further include a controller for monitoring receipt of messages from the message queues by the procedural program message queues, monitoring for calls from the procedural program, and sending the proper message to the calling procedural program. The controller may also timestamp the incoming messages, so that the first-received message is provided to the calling procedural program if more than one message is waiting for it.

Cohen also provides a method for providing messages to procedural window procedures in a computing environment. The method includes determining whether a message for one of the procedural window procedures has been sent from the operating system and storing of the sent message. The method further includes waiting for the procedural window procedure, which corresponds to the stored message, to issue a call for a waiting message; and providing the stored message to the calling procedural window procedure in response to the call.

In summary, Cohen teaches a system and method for making procedures written in procedural languages to respond reactively to messages from event-driven interfaces. It teaches a plurality of message queues for receiving messages from the operating system, each message queue corresponding to each procedural program accessible by the user interface issuing the messages. It also teaches a procedural program interface for determining whether a message is present in one of the message queues and providing the message to the procedural program following receipt of a call from the procedural program. Nowhere does Cohen teach individual context that is created with one or more threads. Nor does Cohen teach that each of the threads can access messages in the context message queue. Nowhere does Cohen teach that threads within a particular context can send messages to other threads within the same context without utilizing the message queue. Nor does Cohen teach that a user interface thread in one context may post messages to a queue associated with a second context.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

Summary of Gomes

Gomes aims to reduce the locking overhead associated with a computing system, thereby improving the performance of the computing system. Gomes purportedly teaches enabling a thread to effectively possess an object without actually owning the object. Gomes allows an object to be speculatively locked such that a thread may operate on the object without acquiring the actual lock associated with the object.

Gomes teaches a method for acquiring the use of an object using a current thread. The method includes determining whether a first bit associated with the object is set to indicate that the object is speculatively owned by a speculative owner thread. If the answer is yes, the speculative owner thread is allowed to use the object without locking the object. The method further includes checking a stored thread ID that is associated with the object and identifies the speculative owner thread, as well as determining whether the stored thread ID identifies the current thread. If the stored thread ID identifies the current thread, the current thread already has speculative use of the object; otherwise, the method further includes locking the object using the speculative owner thread.

Gomes further teaches a method for acquiring ownership of an object in a threaded system. The method includes assigning non-locked ownership, e.g., speculative ownership, of the object to a first thread. This enables the first thread to operate on the object without possessing an ownership lock associated with the object. The method further includes a second thread attempting to obtain the ownership lock associated with the object. This attempt causes the first thread to acquire possession of the ownership lock.

In summary, Gomes teaches enabling a thread associated with an object to effectively possess an object without actually owning the object. Such a thread is called a speculative owner thread. Gomes further teaches that in the case that two threads are competing for the use of the object, the speculative owner thread will get the ownership lock associated with the object. Nowhere does Gomes teach a method for sending a message via a message queue associated with a context. Neither does Gomes teach sending a message from one user interface thread to another user interface thread after determining that another user interface thread is associated with the same context, bypassing the message queue associated with the context. Nor does Gomes teach that a user interface thread in one context may post messages to a queue associated with a second context.

Summary of Allen

Allen purportedly provides an asynchronous inter-process communications arrangement that may be grafted onto an operating system. The asynchronous inter-process communications

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

arrangement comprises a synchronous communication path segment for communicating with a first process, such as a message-sending process, of a pair of communicating processes within a single processor. The arrangement also comprises an asynchronous communications path segment for communicating with the second process of the pair, such as a message-receiving process. The arrangement further comprises a component, such as a procedure, for interconnecting the synchronous and the asynchronous communications path segments to form a communications path extending between the processes of the pair.

The asynchronous segment for communicating with the first process, such as a message-sending process, is semaphore-based. It utilizes a buffer implemented in shared memory and shared by the plurality of message-sending processes. Access to the buffer by the sending processes and the interconnecting procedure is serialized by a semaphore having three states or values.

The asynchronous segment for communicating with the second process of the pair, such as a message-receiving process, is buffer-based. It utilizes circular buffers implemented in shared memory, a different buffer for each message-receiving process. A write pointer to each buffer is controlled by the interconnecting procedure, while a read pointer to each buffer is controlled by the corresponding message-receiving process. The interconnecting procedure notifies the message-receiving process of the presence of a message via a signal facility.

In summary, Allen purportedly provides an asynchronous inter-process communications arrangement that may be grafted onto an operating system. Allen teaches message-sending processes sending messages to a semaphore-controlled buffer, which is shared by a plurality of message-sending processes. Allen further teaches message-receiving processes, each of which is associated with a buffer, which is different from the semaphore-controlled buffer shared by multiple message-sending processes. Nowhere does Allen teaches a context that is associated with one or more user interface threads. Nor does Allen teach allowing user interface threads associated with the same context to communicate with each other, bypassing the message queue associated with the context.

Re: Claim Rejections Under 35 U.S.C. § 112

The Office Action rejected Claims 6, 7, 11, and 12 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter that applicant regards as the invention. These claims are multiple-dependent claims drafted in accordance with the directions outlined in MPEP § 608.01(n) about multiple-dependent claims. The Office Action rejected these claims on the grounds that the claim dependency of Claims 6, 7, 11, and 12 is unclear and makes the claims indefinite. To

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

address the concerns reflected by the rejection of the Office Action on these claims, applicant cancels Claims 6, 7, 11, and 12, and adds Claims 23-27, 28-32, 33-35, and 36-38. These four additional claim sets each correspond to the claim scope outlined by Claims 6, 7, 11, and 12, respectively.

Re: Claim Rejections Under 35 U.S.C. § 103(a)

Independent Claim 1

The Office Action rejected Claim 1 under 35 U.S.C. § 103(a) as being unpatentable over Cohen in view of Gomes.

At its present form, Claim 1 recites:

> 1. A method for sending a message via a high-performance message queue, comprising:
>
> providing a message queue associated with a context;
>
> executing a user interface thread associated with said context;
>
> receiving a request from said user interface thread to send a message to a second user interface thread;
>
> determining whether said second user interface thread is associated with said context; and
>
> in response to determining that said second user interface thread is associated with said context, sending said message from said user interface thread directly to said second user interface thread, thereby bypassing said message queue.

Cohen and Gomes, taken alone or combined, do not teach the subject matter recited by Claim 1. For example, nowhere does Cohen teach a user interface thread associated with a context. In computer programming, a thread generally is a sub-process that is part of a larger process or program. The present application recites that a context, i.e., a user interface component, is created with one or more threads, and that only threads associated with a context may modify an object associated with the context or processing pending messages associated with the context. Cohen does not teach threads. Rather, it teaches window procedures. A window procedure, in the words of Cohen, is typically a C language program or routine resident in memory and designed to provide a set of functions relative to one window available for manipulation by a user. The window procedure can be a word processing program, a

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

spreadsheet program, etc. See Cohen, Col. 4, lines 3-8. Nowhere does Cohen teach executing a user interface thread associated with a context.

Because Cohen does not teach one or more threads associated with a context, it does not teach receiving a request from a first user interface thread to send a message to a second user interface thread associated with the same context, another limitation recited by Claim 1.

The Office Action correctly concludes that Cohen is silent with reference to "determining whether said second user interface thread is associated with said context; and in response to determining that said second user interface thread is associated with said context, sending said message from said user interface thread directly to said second user interface thread, thereby bypassing said message queue." See Office Action, page 5. However, the Office Action alleges that Gomes makes up this deficiency. Applicant respectfully disagrees. Gomes may have suggested different threads associated with the same object; however, nowhere does Gomes teach sending a message from one thread directly to another thread, thereby bypassing the message queue associated with the context. Applicant cannot find any pertinent subject matter in the portions of text of Gomes (Col. 3, lines 16-29; Col. 7, lines 12-39) cited by the Office Action. In these portions of text, Gomes teaches comparing a stored thread ID with the thread ID of the current thread in order to determine whether the stored thread ID and the thread ID of the current thread are substantially identical. If the stored thread ID and the thread ID of the current thread match, then the current thread has speculatively locked the object, meaning the current thread is allowed to use the object as if it owned the object without actually locking the object. See Gomes, Col. 7, lines 27-40. Therefore, nowhere does Gomes teach sending messages from one thread directly to another thread that is associated with the same context, therefore bypassing the message queue of the context.

As a result, neither Cohen nor Gomes, taken alone or in combination, teaches the subject matter recited by Claim 1. Therefore, applicant respectfully requests that Claim 1 is allowable.

Since Claims 2-5 depend from Claim 1, these claims are submitted to be allowable for at least the same reasons that Claim 1 is allowable. Further, these claims are submitted to be allowable for additional reasons.

Dependent Claims 2-5

The Office Action rejected Claims 2-5 under 35 U.S.C. § 103(a) as being unpatentable over Cohen in view of Gomes as applied to Claim 1, and further in view of Allen.

Dependent Claim 2

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

Claim 2 depends from Claim 1, and recites atomically adding a message to the message queue of a second context upon determining that the second user interface thread is not associated with a first context. The Office Action alleges that Gomes teaches the method of Claim 1, and one portion of Claim 2, i.e., "in response to determining that said second user interface thread is not associated with said context." The Office Action further alleges that Allen teaches the other portion of Claim 2, i.e., "atomically adding a message to a queue associated with a second context." The Office Action concludes that it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Allen and Cohen. Applicant respectfully disagrees.

First, the Office Action rejects Claim 2 based on the teaching of Gomes and Allen, yet asserts that it would have been obvious for one of ordinary skill in the art at the time the invention was made to combine the teaching of Allen and Cohen. See Office Action, page 6, paragraph 21. Applicant assumes that this discrepancy is a typographical error and that the Office Action meant to combine the teachings of Gomes and Allen.

Applicant submits that the assertion in the Office Action regarding the obviousness of combining the teachings of Gomes and Allen is based on impermissible hindsight construction of the claimed invention. As noted above, Gomes is directed to enabling a thread to effectively possess an object without actually owning the object. Allen is directed to providing an asynchronous inter-process communications arrangement that can be grafted onto an operating system without adversely affecting any of the operating system's conventional interfaces and operational characteristics. Applicant submits that one of ordinary skill in the art would not be motivated to combine the Gomes teaching with the Allen teaching. Accordingly, applicant submits that there is no motivation for one of ordinary skill in the art to combine the teachings of these two references.

Applicant further submits that the rejection of Claim 2 based on Gomes and Allen is predicated on combining prior art references that contain no teaching or suggestion of how the teachings of the cited references could be combined in any manner, much less the manner recited in Claim 2. The subject matter recited by Claim 2 is taught only by the present application, not by the references. The Office Action fails to point out any teaching or suggestion in the two references related to the desirability of combining their individual teachings. The rejection uses hindsight reasoning based on the present disclosure to "produce" the claimed invention.

Therefore, applicant respectfully submits that Claim 2 is clearly allowable for reasons in addition to the reasons why Claim 1, from which Claim 2 depends, is allowable.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

Dependent Claim 5

Claim 5 depends from Claim 4. Claim 5 recites that atomically adding a message to a queue associated with a second context comprises locking the message and atomically adding the message to a singly-linked list associated with the second context. Gomes and Allen combined do not teach the subject matter recited by Claim 5. In the portion of text of Gomes (Col. 3, lines 25-29) cited by the Office Action, Gomes teaches locking the object using the speculative owner thread when the stored thread ID does not match the current thread ID. Nowhere does Gomes teach locking the message. Further, Claim 5 recites locking a message, instead of locking a message queue, as suggested by the Office Action on page 7, paragraph 24, of the Office Action.

The Office Action further alleges that Cohen teaches adding the message to a singly-linked list associated with the context. In the portion of text of Cohen (Col. 4, lines 13-16) cited by the Office Action, Cohen teaches that the graphics engine causes a message to be placed in its corresponding application queue. Nowhere does Cohen teach a singly-linked list. In computer science, a queue is a singly-linked list, but a singly-linked list is more than a queue. A singly-linked list can also be a stack. A singly-linked list has pointers linked to data. The data itself contains only one pointer to link to another data, which is either the next data or the previous data. If the pointer points to the next data, this singly-linked list is a queue. If the pointer points to a previous data, this singly-linked list is a stack. Even though Cohen teaches an application queue, Cohen does not teach a stack. Therefore, Cohen does not teach a singly-linked list.

Furthermore, there is no teaching or suggestion in either Cohen or Gomes of how the teachings of the two references could be combined in any manner, much less in the manner recited by Claim 5. The Office Action fails to point out any teaching or suggestions in the references related to the desirability of combining their individual teachings. As a result, applicant respectfully submits that Claim 5 is clearly allowable for reasons in addition to the reasons why Claims 4 and 1, from which Claim 5 depends, are allowable.

Claims 23-27 are directed to a computer apparatus for performing the method outlined in Claims 1-5. Claims 28-32 are directed to a computer-readable medium having computer-executable instructions for executing the method outlined in Claims 1-5. Therefore, Claims 23-27 and 28-32 are allowable for the same reasons that Claims 1-5 are allowable.

Independent Claim 8

The Office Action rejected Claim 8 under 35 U.S.C. § 103(a) as being unpatentable over Cohen in view of Allen.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

At its present form, Claim 8 recites:

> 8. A method for posting a message via a high-performance message queue, comprising:
>
> providing a message queue associated with a context;
>
> executing a user interface thread associated with said context;
>
> receiving a request from said user interface thread to post a message to a queue associated with a second context;
>
> atomically adding said message to said queue associated with said second context; and
>
> atomically providing an indication to said second context that a message has been posted to said queue associated with said second context.

Cohen and Allen, taken together or alone, do not teach the subject matter recited by Claim 8. As pointed out above in the discussion regarding Claim 1, Cohen does not teach the use of threads. Therefore, Cohen does not teach executing a user interface thread associated with a context, or receiving a request from the user interface thread to post a message to a queue associated with a second context.

The Office Action correctly concludes that Cohen does not teach atomically adding the message to the queue associated with the second context or atomically providing an indication to the second context that a message has been posted to the queue associated with the second context. The Office Action alleges that Allen makes up this deficiency. Applicant respectfully disagrees.

First, the teachings of Allen and Cohen are too distinctly different to consider that it would be obvious for one of ordinary skill in the art at the time the invention was made to combine the teachings of Allen and Cohen. As pointed out above, Cohen is directed to enabling programs written in procedural languages to respond reactively to messages from event-driven interfaces. Allen is directed to providing an asynchronous inter-process communications arrangement that can be grafted onto the operating system. Applicant submits that one of ordinary skill in the art would not be motivated to combine the teaching of Allen with the teaching of Cohen. Further, there is no teaching or suggestion in Cohen or Allen, taken alone or in combination, why it would be obvious to combine the individual teachings of these references. More importantly, even if these references were combinable--which applicant categorically denies--the resulting combination would not anticipate the subject matter of Claim 8. As noted

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

above, Cohen does not teach the subject matter recited by Claim 8, because Cohen does not teach a user interface thread, atomically adding messages to queues associated with a second context, or atomically providing an indication to the second context that a message has been posted to the queue associated with the second context. Nor does Allen teach the subject matter of Claim 8, for Allen at least fails to teach a user interface thread associated with a context. Therefore, applicant respectfully submits that Cohen and Allen, taken alone or combined, do not teach the subject matter recited by Claim 8, and that Claim 8 is clearly allowable.

Since Claims 9-10 depend from Claim 8, these claims are submitted to be allowable for at least the same reasons that Claim 8 is allowable. Further, these claims are submitted to be allowable for additional reasons.

Dependent Claim 9

In its present form, Claim 9 recites validating parameters associated with a message, determining a processing function to handle the dequeueing of the message, and completing a message entry for the message including the validated parameters and the identity of the processing function. The Office Action alleges that Cohen teaches the subject matter recited by Claim 9 in that Cohen teaches forwarding a message to the appropriate window procedure. See Office Action, page 8, paragraph 30. Applicant respectfully disagrees. The Office Action fails to point out how Cohen specifically teaches the limitations recited by Claim 9. Nowhere does Cohen teach specifically validating parameters associated with the message, determining a processing function to handle the dequeueing of the message, and completing a message entry for the message including the validate parameters and the identity of the processing function. Therefore, Claim 9 is allowable for these reasons in addition to the reasons that Claim 8, from which Claim 9 depends, is allowable.

Dependent Claim 10

Claim 10 depends from Claim 9 and recites locking a message entry and atomically adding the message entry to a singly-linked list associated with a context. The Office Action rejected Claim 10 under 35 U.S.C. § 103(a) as being unpatentable over Cohen in view of Allen as applied to Claim 8, and further in view of Gomes. Applicant respectfully disagrees. As the above discussion regarding Claim 5 shows, Cohen fails to teach a singly-linked list associated with a context. Consequently, Cohen fails to teach adding a message entry to a singly-linked list associated with the context. Neither does Allen teach a singly-linked list associated with a context, either.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

The Office Action correctly concludes that Cohen and Allen combined did not teach a locking system. The Office Action alleged that Gomes makes up this deficiency and teaches a locking system. Even if Gomes does teach the locking system recited by Claim 10, Gomes still does not teach atomically adding a message entry to a singly-linked list associated with a context. Therefore, Cohen, Allen and Gomes, taken alone or in combination, do not teach the subject matter recited by Claim 10.

Furthermore, as pointed out in above discussion regarding Claim 2 and Claim 8, each of the three references is directed to a distinctly different system. There is no teaching or suggestion of how the teachings of the three references could be combined in any manner, much less the manner recited in Claim 10. Applicant submits that one of ordinary skill in the art would not be motivated to combine these teachings. Furthermore, there is no teaching or suggestion in these references, taken alone or in combination, why it would be obvious to combine the individual teachings of these references. More importantly, as noted above, even if it is obvious for one of ordinary skill in the art to modify the teachings of the three references to achieve the claimed limitations in Claim 10--which applicant categorically denies--the combined teachings still would not anticipate the subject matter of Claim 10 when the subject matter of Claim 10 is considered in combination with the subject matter of Claims 9 and 8, from which Claim 10 depends. Consequently, applicant respectfully submits that Claim 10 is allowable for reasons in addition to the reasons why Claims 8 and 9 are allowable.

Claims 33-35 are directed to a computer apparatus for performing the method outlined in Claims 8-10. Claims 36-38 are directed to a computer-readable medium having computer-executable instructions for executing the method outlined in Claims 8-10. Therefore, Claims 33-35 and 36-38 are allowable for the same reasons that Claims 8-10 are allowable.

CONCLUSION

In view of the foregoing comments, applicant respectfully submits that all of the claims in this application are clearly allowable in view of the cited and applied references. Consequently, early and favorable action allowing these claims and passing this application to

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

issue is respectfully solicited. If the Examiner has any questions or comments concerning this application, the Examiner is invited to contact the applicant's undersigned attorney at the number below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS[PLLC]

Joy Y. Xiang
Registration No. 55,747
Direct Dial No. 206.695.1607

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid and addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date.

Date: 10/1/04

JYX:dmg/nfs

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS[PLLC]
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100